

The specified file is not just opened and read from a static location. Instead, a whole subrequest is processed by Nginx, and the body of the response is inserted instead of the `include` tag.

## Conditional structure

The Rewrite module introduces a new set of directives and blocks among which is the `if` conditional structure:

```
server {
    if ($request_method = POST) {
        [...]
    }
}
```

This allows you to apply a configuration according to the specified condition. If the condition is true, the configuration is applied; otherwise, it isn't.

The following table describes the various syntaxes accepted when forming a condition:

Operator	Description
None	The condition is true if the specified variable or data is not equal to an empty string or a string starting with the character 0: <pre>if (\$string) {     [...] }</pre>
<code>=, !=</code>	The condition is true if the argument preceding the <code>=</code> symbol is equal to the argument following it. The following example can be read as "if the <code>request_method</code> is equal to <code>POST</code> , then apply the configuration": <pre>if (\$request_method = POST) {     [...] }</pre> <p>The <code>!=</code> operator does the opposite: "if the request method is not equal to <code>GET</code>, then apply the configuration":</p> <pre>if (\$request_method != GET) {     [...] }</pre>

Operator	Description
<code>~, ~*, !~, !~*</code>	<p>The condition is true if the argument preceding the <code>~</code> symbol matches the regular expression pattern placed after it:</p> <pre>if (\$request_filename ~ "\.txt\$") {     [...] }</pre> <p><code>~</code> is case-sensitive, <code>~*</code> is case-insensitive. Use the <code>!</code> symbol to negate the matching:</p> <pre>if (\$request_filename !~* "\.php\$") {     [...] }</pre> <p>Note that you can insert the capture buffers in the regular expression:</p> <pre>if (\$uri ~ "^/search/(.*)\$") {     set \$query \$1;     rewrite ^ http://google.com/search?q=\$query; }</pre>
<code>-f, !-f</code>	<p>Tests the existence of the specified file:</p> <pre>if (-f \$request_filename) {     [...] # if the file exists }</pre> <p>Use <code>!-f</code> to test the non-existence of the file:</p> <pre>if (!-f \$request_filename) {     [...] # if the file does not exist }</pre>
<code>-d, !-d</code>	Similar to the <code>-f</code> operator, is used for testing the existence of a directory.
<code>-e, !-e</code>	Similar to the <code>-f</code> operator, is used for testing the existence of a file, directory, or symbolic link.
<code>-x, !-x</code>	Similar to the <code>-f</code> operator, is used for testing whether a file exists and is executable.

As of version 1.8, there is no `else` or `else if`-like instruction. However, other directives allowing you to control the configuration flow sequencing are available.

You might wonder: what are the advantages of using a `location` block over an `if` block? Indeed, in the following example, both seem to have the same effect:

```
if ($uri ~ /search/) {
    [...]
}
location ~ /search/ {
    [...]
}
```